Fast Fourier Extension with Discrete Prolate Spheroidal Sequences

Roel Matthysen

June 23, 2014

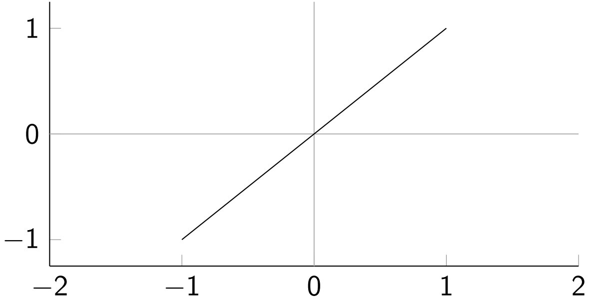
Joint work with Daan Huybrechs
Department of Computer Science, KU Leuven

Fourier Extension

Fourier Extension principle

Constructing an exponentially convergent approximation of a function on a given interval, by fitting a Fourier series on a larger domain.

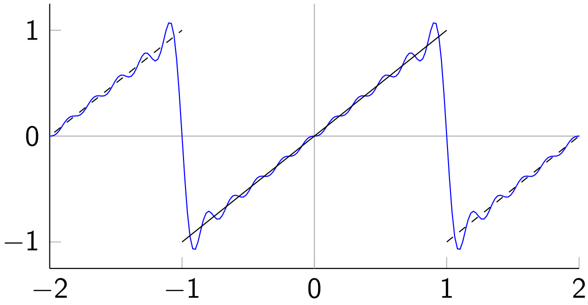
Example $f(x) = x$ on $[-1, 1]$



Fourier Extension

Fourier Series

$$f_N(x) = \frac{a_0}{2} + \sum_{n=1}^{N} \left[a_n \cos 2\pi nx + b_n \sin 2\pi nx\right]$$



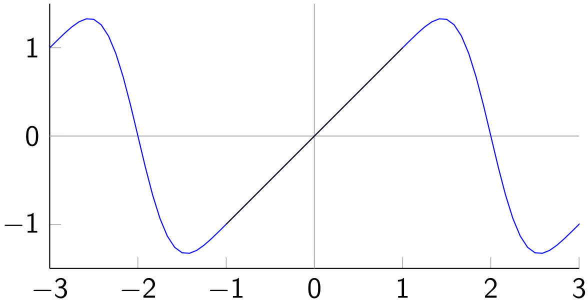
Fourier Extension

Fourier Extension

Fourier series over larger domain $[-T, T]$.

$$g \in G_N : g(x) = \frac{a_0}{2} + \sum_{n=1}^{N} \left[a_n \cos 2\pi \frac{nx}{T} + b_n \sin 2\pi \frac{nx}{T} \right]$$

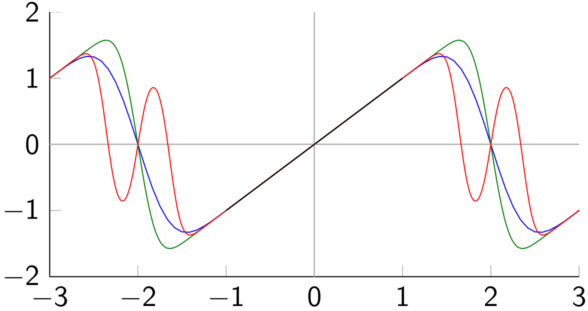
$$g_N := \arg \min_{g \in G_N} ||f - g||_{L^2_{[-1,1]}}$$



Fourier Extension

Redundancy

- Fourier basis restricted to interval constitutes a frame
- Least squares fitting problem has many good approximate solutions



Why Fourier Extensions

- Exponential convergence for smooth but non-periodic functions[1]
- Good resolution power (dof per wavelength) when T is close to 1[2]
- Approximation from uniformly spaced data
- A fast $O(N \log(N))$ algorithm exists[3], using FFTs, for the special case $T = 2$
  - This talk : A fast solver for general T

---

[1] **Huybrechs 2010; Adcock**, **Huybrechs**, **Martin-Vaquero 2014**.

[2] **Adcock**, **Huybrechs 2011**.

[3] **Lyon 2012**.

Fourier Extension

Notation

- Frame basis functions $\phi_n(x) = \frac{1}{\sqrt{2T}} e^{i\frac{\pi n}{T}x}, n = -N, \ldots, N$
- Approximation $g(x) = \sum_{k=-N}^{N} a_k \phi_k(x)$

Fitting the Fourier Extension

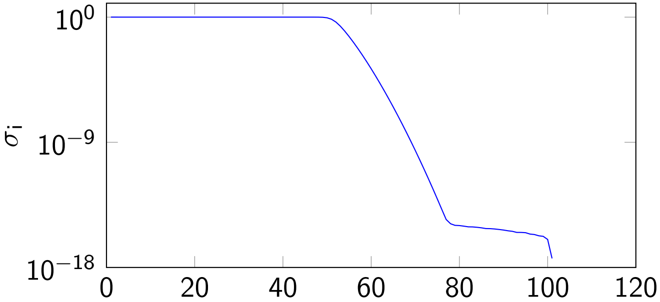
- Collocation at equidistant points $x_j = \frac{j}{M}, j = -M, \ldots, M$.
- Least squares problem $A_{ij} = \phi_i(x_j)$, $b_j = f(x_j)$

$$Aa \approx b$$

- Some oversampling $M = \eta N$ required, A is rectangular
- Goal: A fast solver for this system

A closer look at A

Singular values of A



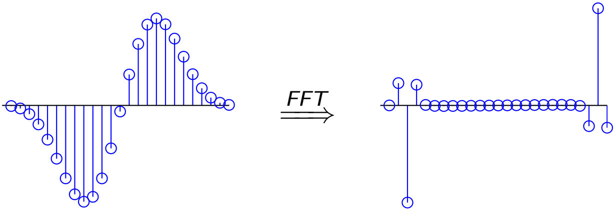
- Least-squares system $Aa = b$ is extremely ill-conditioned, $\kappa(A) = \mathcal{O}(\epsilon^{-1})$
- Solving with Truncated SVD or MATLAB backslash still gives approximation to machine accuracy[4]

[4] **Adcock**, **Huybrechs**, **Martin-Vaquero 2014**.

A closer look at A

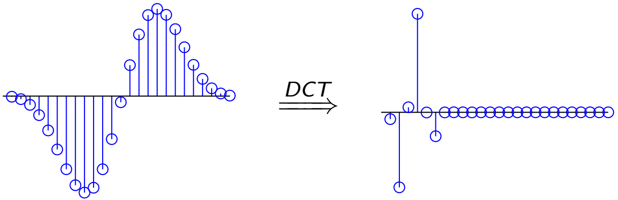
Singular vectors of A

- Are Periodic Discrete Prolate Spheroidal Sequences
- Corresponding singular values:
  - $O(N/T)$ are in $[1-\epsilon, 1)$
  - $O(N-N/T)$ are in $(0, \epsilon]$
  - $O(\log N)$ "intermediate" values in $(\epsilon, 1-\epsilon)$
- Easily obtained as eigenvectors of tridiagonal matrix
- Optimal frequency localisation w.r.t. FFT



Separating the singular values

- Frequency localization of singular vectors (left & right), energy in first N/T bins is proportional to σ_i



- $A = USV^*$, C=DCT-matrix

$$CU \approx \begin{bmatrix} F_1 & \epsilon \\ \epsilon & F_2 \end{bmatrix}$$

$$CA \approx \begin{bmatrix} F_1 & \epsilon \\ \epsilon & F_2 \end{bmatrix} \begin{bmatrix} S_1 V_1^* \\ S_2 V_2^* \end{bmatrix} = \begin{bmatrix} F_1 S_1 V_1^* + O(\epsilon) \\ F_1 S_2 V_2^* + O(\epsilon) \end{bmatrix}$$

Separating the singular values

System CAa=Cb

- $CA = \begin{bmatrix} \overline{A}_1 \\ \overline{A}_2 \end{bmatrix}$, $Cb = \begin{bmatrix} \overline{b}_1 \\ \overline{b}_2 \end{bmatrix}$, where
  - $\kappa(\overline{A}_1) \approx 1$, full rank
  - $\kappa(\overline{A}_2) \approx \epsilon^{-1}$, $rank(\overline{A}_2) = O(\log N)$

Algorithm using random matrices

1. Solve $\overline{A}_1 a_1 = \overline{b}_1$ iteratively
2. Solve $\overline{A}_2 a_2 = \overline{b}_2 - \overline{A}_2 a_1$, with $a_2 \in null(\overline{A}_1)$
   - Solve $\overline{A}_1 q = \overline{A}_1 r$, for $O(\log N)$ random vectors r
     - $r - q$ is in $null(\overline{A}_1)$
   - Construct reduced system $L = \overline{A}_2(R - Q)$
   - Solve $Lz = \overline{b}_2 - \overline{A}_2 a_1$ with an SVD
   - Expand z : $a_2 = (R - Q)z$
3. $a = a_1 + a_2$

Separating the singular values

System CAa=Cb

- $CA = \begin{bmatrix} \overline{A}_1 \\ \overline{A}_2 \end{bmatrix}$, $Cb = \begin{bmatrix} \overline{b}_1 \\ \overline{b}_2 \end{bmatrix}$, where
  - $\kappa(\overline{A}_1) \approx 1$, full rank
  - $\kappa(\overline{A}_2) \approx \epsilon^{-1}$, $rank(\overline{A}_2) = O(\log N)$

Algorithm using random matrices

1. Solve $\overline{A}_1 a_1 = \overline{b}_1$ iteratively
2. Solve $\overline{A}_2 a_2 = \overline{b}_2 - \overline{A}_2 a_1$, with $a_2 \in null(\overline{A}_1)$
   - Solve $\overline{A}_1 q = \overline{A}_1 r$, for $O(\log N)$ random vectors r
     - $r - q$ is in $null(\overline{A}_1)$
   - Construct reduced system $L = \overline{A}_2(R - Q)$
   - Solve $Lz = \overline{b}_2 - \overline{A}_2 a_1$ with an SVD
   - Expand z : $a_2 = (R - Q)z$
3. $a = a_1 + a_2$

Fast Fourier Extension with Discrete Prolate Spheroidal Sequences

Roel Matthysen

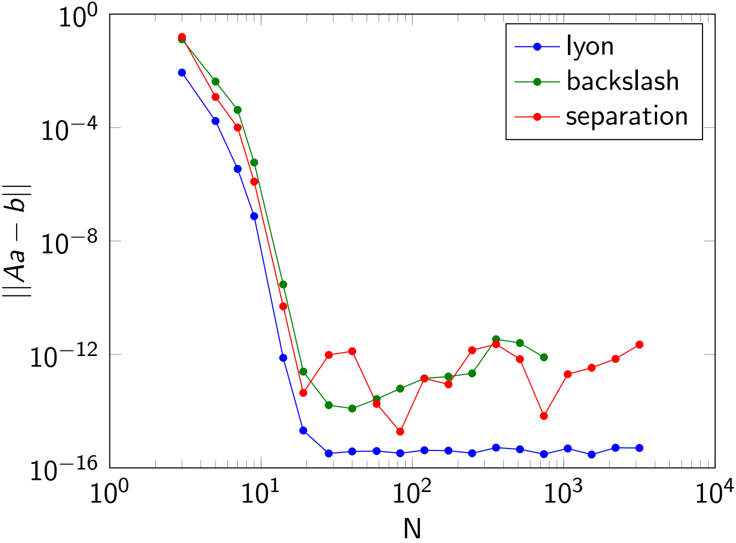
Fourier Extension

A closer look at A

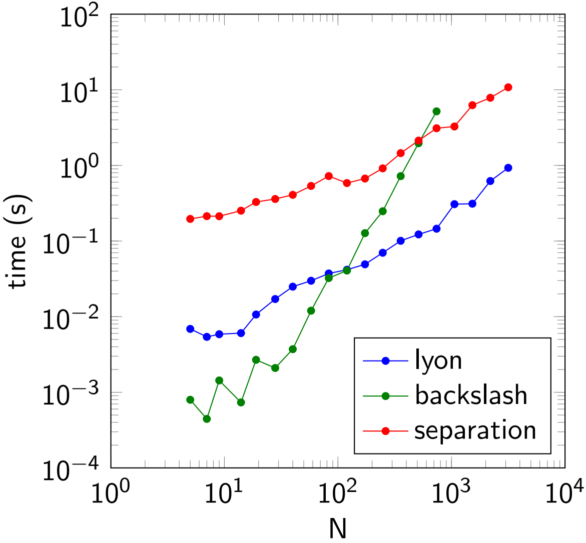
Algorithm for general T

Conclusions

Results



Results


10^2
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
time (s)
10^0
10^1
10^2
10^3
10^4
N
lyon
backslash
separation

Conclusions

- Separation algorithm on par with M. Lyon algorithm on complexity and convergence
- Constant factor slower in practice, but more flexible
  - Possible to vary T with M
- P-DPSS provide theoretical framework for the FE method for equispaced data